Optimization-based motion planning

SIDRA Summer School,
Bertinoro 2023

Paolo Falcone

Department of Electrical Engineering,
Chalmers University of Technology,
Gothenburg, Sweden

Dipartimento di Ingegneria "Enzo Ferrari",
Università di Modena e Reggio Emilia,
Modena

Lecture objectives

- Optimization-based motion planning problem statement

Lecture objectives

- Optimization-based motion planning problem statement
- Model Predictive Control

Lecture objectives

- Optimization-based motion planning problem statement
- Model Predictive Control
- From MPC to nonlinear or linear/quadratic programming

Lecture objectives

- Optimization-based motion planning problem statement
- Model Predictive Control
- From MPC to nonlinear or linear/quadratic programming
- Step-by-step preparation of the motion planning problem
    1. Space vs. Time-based problem formulations

Lecture objectives

- Optimization-based motion planning problem statement
- Model Predictive Control
- From MPC to nonlinear or linear/quadratic programming
- Step-by-step preparation of the motion planning problem
  1. Space vs. Time-based problem formulations
  2. Cost design

Lecture objectives

- Optimization-based motion planning problem statement
- Model Predictive Control
- From MPC to nonlinear or linear/quadratic programming
- Step-by-step preparation of the motion planning problem
  1. Space vs. Time-based problem formulations
  2. Cost design
  3. Reference path

Lecture objectives

- Optimization-based motion planning problem statement
- Model Predictive Control
- From MPC to nonlinear or linear/quadratic programming
- Step-by-step preparation of the motion planning problem
    1. Space vs. Time-based problem formulations
    2. Cost design
    3. Reference path
    4. Vehicle modeling

Lecture objectives

- Optimization-based motion planning problem statement
- Model Predictive Control
- From MPC to nonlinear or linear/quadratic programming
- Step-by-step preparation of the motion planning problem
  1. Space vs. Time-based problem formulations
  2. Cost design
  3. Reference path
  4. Vehicle modeling
  5. Safety constraints. Static obstacles

Problem statement

Find the trajectory to track (path to follow) as the solution of the problem of minimizing a desired cost, while satisfying physical and design constraints.

Problem statement

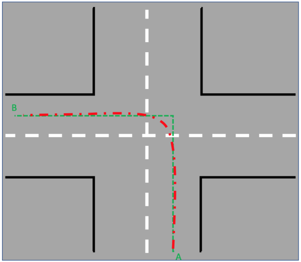
Find the trajectory to track (path to follow) as the solution of the problem of minimizing a desired cost, while satisfying physical and design constraints.

Designing the cost.

Problem statement

Find the trajectory to track (path to follow) as the solution of the problem of minimizing a desired cost, while satisfying physical and design constraints.

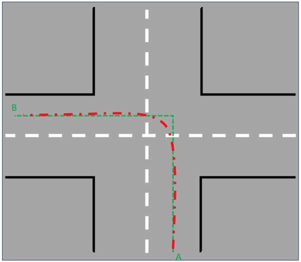
Designing the cost. In road transportation applications a "reference" path is likely to be available. E.g., the lane centerline of the *desired route*,

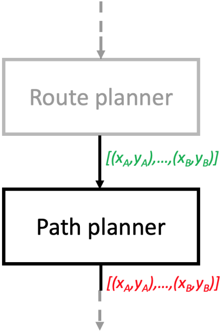


Problem statement

Find the trajectory to track (path to follow) as the solution of the problem of minimizing a desired cost, while satisfying physical and design constraints.

Designing the cost. In road transportation applications a “reference” path is likely to be available. E.g., the lane centerline of the *desired route*, which can be assumed to be given by a *route planner*.

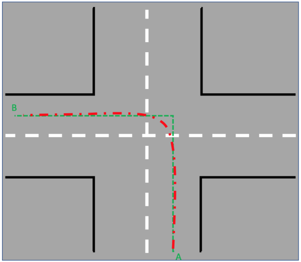


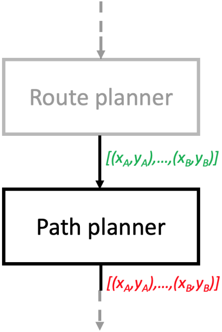


Problem statement

Find the trajectory to track (path to follow) as the solution of the problem of minimizing a desired cost, while satisfying physical and design constraints.

Designing the cost. In road transportation applications a "reference" path is likely to be available. E.g., the lane centerline of the *desired route*, which can be assumed to be given by a *route planner*.





The cost should then be designed such that the *planned path* minimizes, in some sense, the deviation from the *reference path*.

Optimization-based motion planning problem formulation

The optimization-based motion planning reads as

Optimization-based motion planning problem formulation

The optimization-based motion planning reads as

$$\underset{\textit{path/trajectory}}{\text{minimize}} \qquad \text{deviation from reference path/trajectory}$$

Optimization-based motion planning problem formulation

The optimization-based motion planning reads as

$$\begin{aligned} \underset{\textit{path/trajectory}}{\text{minimize}} \quad & \text{deviation from reference path/trajectory} \\ \text{subject to} \quad & \\ & \text{vehicle model} \end{aligned}$$

Optimization-based motion planning problem formulation

The optimization-based motion planning reads as

$$
\begin{aligned}
\underset{\textit{path/trajectory}}{\text{minimize}} \quad & \text{deviation from reference path/trajectory} \\
\text{subject to} \quad & \\
& \text{vehicle model} \\
& \text{safety constraints}
\end{aligned}
$$

Optimization-based motion planning problem formulation

The optimization-based motion planning reads as

$$\begin{aligned} \underset{\textit{path/trajectory}}{\text{minimize}} \quad & \text{deviation from reference path/trajectory} \\ \text{subject to} \quad & \\ & \text{vehicle model} \\ & \text{safety constraints} \\ & \text{actuator limitations} \end{aligned}$$

Optimization-based motion planning problem formulation

The optimization-based motion planning reads as

$$
\begin{array}{ll}
\underset{\textit{path/trajectory}}{\text{minimize}} & \text{deviation from reference path/trajectory} \\
\text{subject to} & \\
 & \text{vehicle model} \\
 & \text{safety constraints} \\
 & \text{actuator limitations} \\
 & \text{design (e.g., comfort) constraints}
\end{array}
$$

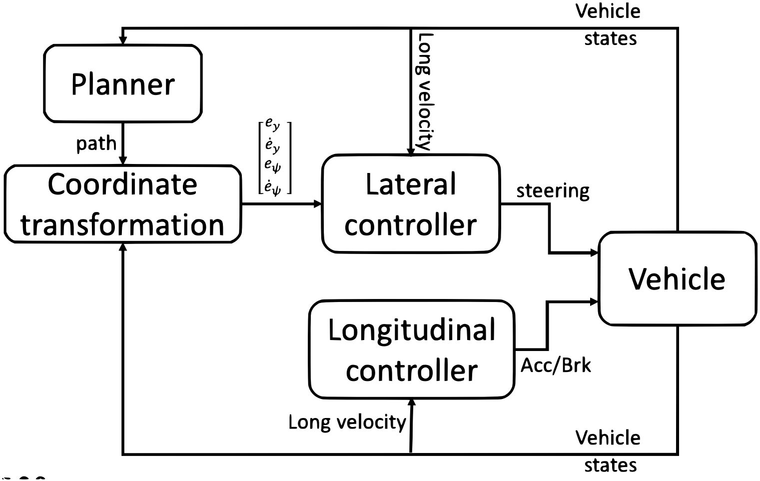
Optimization-based motion planning problem formulation

The optimization-based motion planning reads as

$$
\begin{aligned}
\underset{\textit{path/trajectory}}{\text{minimize}} \quad & \text{deviation from reference path/trajectory} \\
\text{subject to} \quad & \\
& \text{vehicle model} \\
& \text{safety constraints} \\
& \text{actuator limitations} \\
& \text{design (e.g., comfort) constraints}
\end{aligned}
$$

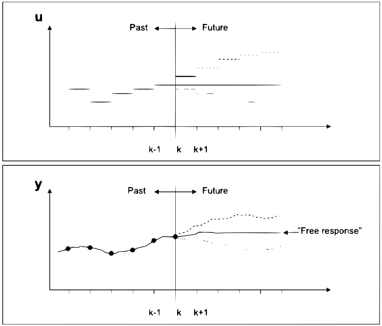
The planned motion is periodically updated based on the current vehicle and environment state, and sent to motion control layer.

Architecture



The receding horizon idea

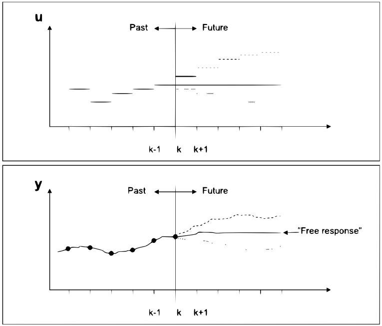
The core of the MPC approach, the receding horizon idea:



The receding horizon idea

The core of the MPC approach, the receding horizon idea:

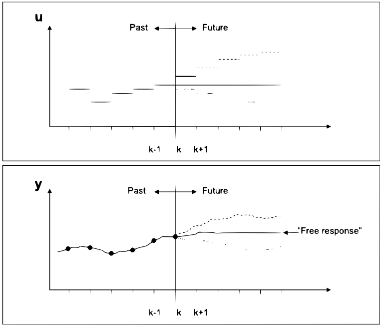
1. At time instant k, *predict* the process response over a finite *prediction horizon* N; this response depends on the sequence of future control inputs over the *control horizon* M.



The receding horizon idea

The core of the MPC approach, the receding horizon idea:

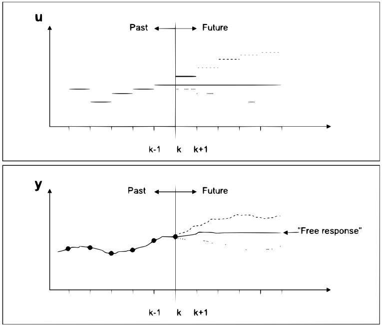
1. At time instant k, *predict* the process response over a finite *prediction horizon* N; this response depends on the sequence of future control inputs over the *control horizon* M.
2. Pick the control sequence which gives the best performance in terms of a specified *objective*, *cost function* or *criterion*.



The receding horizon idea

The core of the MPC approach, the receding horizon idea:

1. At time instant k, *predict* the process response over a finite *prediction horizon* N; this response depends on the sequence of future control inputs over the *control horizon* M.
2. Pick the control sequence which gives the best performance in terms of a specified *objective*, *cost function* or *criterion*.
3. Apply the first element in the control sequence to the process, discard the rest of the sequence, and return to step 1.



The Receding Horizon recipe

The MPC recipe for the example:

1. At time k, *predict* the output N samples ahead:

$$\hat{y}(k+1|k), \ldots, \hat{y}(k+N|k)$$

The Receding Horizon recipe

The MPC recipe for the example:

1. At time k, *predict* the output N samples ahead:

$$\hat{y}(k+1|k), \ldots, \hat{y}(k+N|k)$$

2. The predictions depend on future control inputs

$$\hat{u}(k|k), \hat{u}(k+1|k), \ldots, \hat{u}(k+M-1|k)$$

   (Normally, $M < N$, and we assume that u is either 0 or unchanged after this.)

The Receding Horizon recipe

The MPC recipe for the example:

1. At time k, *predict* the output N samples ahead:

$$\hat{y}(k+1|k), \ldots, \hat{y}(k+N|k)$$

2. The predictions depend on future control inputs

$$\hat{u}(k|k), \hat{u}(k+1|k), \ldots, \hat{u}(k+M-1|k)$$

   (Normally, $M < N$, and we assume that u is either 0 or unchanged after this.)

3. Minimize a criterion (now adopting the index notation : as in Matlab)

$$V(k) = V(\hat{y}(k+1:k+N|k), \hat{u}(k:k+M-1|k))$$

   with respect to the control sequence $\hat{u}(k:k+M-1|k)$

The Receding Horizon recipe

The MPC recipe for the example:

1. At time k, *predict* the output N samples ahead:

$$\hat{y}(k+1|k), \ldots, \hat{y}(k+N|k)$$

2. The predictions depend on future control inputs

$$\hat{u}(k|k), \hat{u}(k+1|k), \ldots, \hat{u}(k+M-1|k)$$

   (Normally, $M < N$, and we assume that u is either 0 or unchanged after this.)

3. Minimize a criterion (now adopting the index notation : as in Matlab)

$$V(k) = V(\hat{y}(k+1:k+N|k), \hat{u}(k:k+M-1|k))$$

   with respect to the control sequence $\hat{u}(k:k+M-1|k)$

4. Apply the first control signal in the sequence to the process:

$$u(k) = \hat{u}(k|k)$$

The Receding Horizon recipe

The MPC recipe for the example:

1. At time k, *predict* the output N samples ahead:

$$\hat{y}(k+1|k), \ldots, \hat{y}(k+N|k)$$

2. The predictions depend on future control inputs

$$\hat{u}(k|k), \hat{u}(k+1|k), \ldots, \hat{u}(k+M-1|k)$$

   (Normally, $M < N$, and we assume that u is either 0 or unchanged after this.)

3. Minimize a criterion (now adopting the index notation : as in Matlab)

$$V(k) = V(\hat{y}(k+1:k+N|k), \hat{u}(k:k+M-1|k))$$

   with respect to the control sequence $\hat{u}(k:k+M-1|k)$

4. Apply the first control signal in the sequence to the process:

$$u(k) = \hat{u}(k|k)$$

5. Increment time $k := k+1$ and go to 1.

MPC ingredients

- An *internal model* describing process and disturbances

MPC ingredients

- An *internal model* describing process and disturbances
- An *estimator/predictor* to determine the evolution of the state

MPC ingredients

- An *internal model* describing process and disturbances
- An *estimator/predictor* to determine the evolution of the state
- An *objective/criterion* to express the desired system behaviour

MPC ingredients

- An *internal model* describing process and disturbances
- An *estimator/predictor* to determine the evolution of the state
- An *objective/criterion* to express the desired system behaviour
- An *online optimization algorithm* to determine future control actions

MPC ingredients

- An *internal model* describing process and disturbances
- An *estimator/predictor* to determine the evolution of the state
- An *objective/criterion* to express the desired system behaviour
- An *online optimization algorithm* to determine future control actions
- The *receding horizon* principle

The RHC problem

Consider the system subject to state and input constraints

$$S: \quad x^+ = f(x,u),\ x \in \mathcal{X},\ u \in \mathcal{U}. \tag{1}$$

The RHC problem

Consider the system subject to state and input constraints

$$S: \quad x^+ = f(x,u),\ x \in \mathcal{X},\ u \in \mathcal{U}. \tag{1}$$

Optimization problem:

$$P: \quad \min_{u(0:N-1)} V_N(x(0), u(0:N-1))$$

where the minimization is with respect to the sequence of control inputs

$$u(0:N-1) = \{u(0), u(1), \ldots, u(N-1)\}$$

and subject to the system model, state and input constraints (1).

The RHC problem

Consider the system subject to state and input constraints

$$S: \quad x^+ = f(x,u),\ x \in \mathcal{X},\ u \in \mathcal{U}. \tag{1}$$

Optimization problem:

$$P: \quad \min_{u(0:N-1)} V_N(x(0), u(0:N-1))$$

where the minimization is with respect to the sequence of control inputs

$$u(0:N-1) = \{u(0), u(1), \ldots, u(N-1)\}$$

and subject to the system model, state and input constraints (1).

The *objective* or *criterion* or *cost function* V_N is given by

$$\begin{aligned} V_N(x(0), u(0:N-1)) &= \sum_{i=0}^{N-1} (x^\top(i) Q x(i) + u^\top(i) R u(i)) + x^\top(N) P_f x(N) \\ &= \sum_{i=0}^{N-1} l(x(i), u(i)) + l_f(x(N)) \end{aligned} \tag{2}$$

The RHC problem

Consider the system subject to state and input constraints

$$S: \quad x^+ = f(x,u),\ x \in \mathcal{X},\ u \in \mathcal{U}. \tag{1}$$

Optimization problem:

$$P: \quad \min_{u(0:N-1)} V_N(x(0), u(0:N-1))$$

where the minimization is with respect to the sequence of control inputs

$$u(0:N-1) = \{u(0), u(1), \ldots, u(N-1)\}$$

and subject to the system model, state and input constraints (1).
The *objective* or *criterion* or *cost function* V_N is given by

$$\begin{aligned} V_N(x(0), u(0:N-1)) &= \sum_{i=0}^{N-1} (x^\top(i) Q x(i) + u^\top(i) R u(i)) + x^\top(N) P_f x(N) \\ &= \sum_{i=0}^{N-1} l(x(i), u(i)) + l_f(x(N)) \end{aligned} \tag{2}$$

Remark 1: All $x(i)$ are functions of $x(0)$ and $u(0:N-1)$ via the model (1)!

The RHC problem

Consider the system subject to state and input constraints

$$S: \quad x^+ = f(x,u),\ x \in \mathcal{X},\ u \in \mathcal{U}. \tag{1}$$

Optimization problem:

$$P: \quad \min_{u(0:N-1)} V_N(x(0), u(0:N-1))$$

where the minimization is with respect to the sequence of control inputs

$$u(0:N-1) = \{u(0), u(1), \ldots, u(N-1)\}$$

and subject to the system model, state and input constraints (1).

The *objective* or *criterion* or *cost function* V_N is given by

$$\begin{aligned} V_N(x(0), u(0:N-1)) &= \sum_{i=0}^{N-1} (x^\top(i) Q x(i) + u^\top(i) R u(i)) + x^\top(N) P_f x(N) \\ &= \sum_{i=0}^{N-1} l(x(i), u(i)) + l_f(x(N)) \end{aligned} \tag{2}$$

Remark 1: All $x(i)$ are functions of $x(0)$ and $u(0:N-1)$ via the model (1)!

Remark 2: The first term $x^\top(0) Q x(0)$ in the objective is really redundant but is kept for notational convenience.

Motion planning. The optimization problem

Let $\mathcal{X} = \{x : \; g_x(x) \le 0\}$, $\mathcal{U} = \{u : \; g_u(u) \le 0\}$.

Motion planning. The optimization problem

Let $\mathcal{X} = \{x : \; g_x(x) \leq 0\}$, $\mathcal{U} = \{u : \; g_u(u) \leq 0\}$. Every time instant solve the problem

$$\begin{aligned}
&\min_{u(0:N-1),\; x(0:N)} V_N(x(0), u(0\!:\!N-1)) \\
&\text{subject to} \\
&x(1) - f(x(0), u(0)) = 0, \\
&\vdots \\
&x(N) - f(x(N-1), u(N-1)) = 0, \\
&g_x(x(i)) \leq 0, \; i = 1 : N, \\
&g_u(u(i)) \leq 0, \; i = 0 : N-1.
\end{aligned}$$

where the initial state in the prediction model is set equal to $x(t)$.

Motion planning. The optimization problem

Let $\mathcal{X} = \{x : g_x(x) \le 0\}$, $\mathcal{U} = \{u : g_u(u) \le 0\}$. Every time instant solve the problem

$$
\begin{aligned}
&\min_{u(0:N-1),\ x(0:N)} V_N(x(0), u(0:N-1)) \\
&\text{subject to} \\
&x(1) - f(x(0), u(0)) = 0, \\
&\vdots \\
&x(N) - f(x(N-1), u(N-1)) = 0, \\
&g_x(x(i)) \le 0,\ i = 1 : N, \\
&g_u(u(i)) \le 0,\ i = 0 : N-1.
\end{aligned}
$$

where the initial state in the prediction model is set equal to $x(t)$.

In Matlab

```
[x,fval,exitflag,output]=fmincon(V_N,x0,[],[],[],[],[],[],g_x_g_u)
```

Motion planning. The optimization problem

Let $\mathcal{X} = \{x : \ g_x(x) \leq 0\}$, $\mathcal{U} = \{u : \ g_u(u) \leq 0\}$. Every time instant solve the problem

$$
\begin{aligned}
&\min_{u(0:N-1),\ x(0:N)} V_N(x(0), u(0\!:\!N-1)) \\
&\text{subject to} \\
&x(1) - f(x(0), u(0)) = 0, \\
&\vdots \\
&x(N) - f(x(N-1), u(N-1)) = 0, \\
&g_x(x(i)) \leq 0, \ i = 1 : N, \\
&g_u(u(i)) \leq 0, \ i = 0 : N-1.
\end{aligned}
$$

where the initial state in the prediction model is set equal to $x(t)$.

In Matlab

```
[x,fval,exitflag,output]=fmincon(V_N,x0,[],[],[],[],[],[],g_x_g_u)
```

Apply to the system the control input $u(t) = u^*(0)$ and repeat the optimization from the next state (over a shifted time horizon).

Motion planning. The optimization problem

If the system is linear ($x^+ = Ax + Bu$), the cost quadratic and the constraints convex

$$\mathcal{X} = \{x : \; A_x \leq b_x\}, \; \mathcal{U} = \{u : \; A_u \leq b_u\}.$$

Motion planning. The optimization problem

If the system is linear ($x^+ = Ax + Bu$), the cost quadratic and the constraints convex

$$\mathcal{X} = \{x : \ A_x \leq b_x\},\ \mathcal{U} = \{u : \ A_u \leq b_u\}.$$

The constrained optimization problem becomes (see the batch approach for LQ)

$$\begin{aligned}
&\min_{\boldsymbol{u}} \boldsymbol{u}^\top(\Gamma^\top \bar{Q}\Gamma + \bar{R})\boldsymbol{u} + 2x^\top(0)\Omega^\top \bar{Q}\Gamma \boldsymbol{u} + x^\top(0)(Q + \Omega^\top \bar{Q}\Omega)x(0) \\
&\text{subject to} \\
&[A_x\ A_x\ \cdots\ A_x](\Omega x(0) + \Gamma \boldsymbol{u}) \leq [b_x\ b_x\ \ldots\ b_x]^T,, \\
&[A_u\ A_u\ \cdots\ A_u]\boldsymbol{u} \leq [b_u\ b_u\ \ldots\ b_u]^T,
\end{aligned}$$

where the initial state in the prediction model is set equal to $x(t)$.

Motion planning. The optimization problem

If the system is linear ($x^+ = Ax + Bu$), the cost quadratic and the constraints convex

$$\mathcal{X} = \{x : \ A_x \leq b_x\}, \ \mathcal{U} = \{u : \ A_u \leq b_u\}.$$

The constrained optimization problem becomes (see the batch approach for LQ)

$$\min_{\boldsymbol{u}} \boldsymbol{u}^\top(\Gamma^\top \bar{Q}\Gamma + \bar{R})\boldsymbol{u} + 2x^\top(0)\Omega^\top \bar{Q}\Gamma \boldsymbol{u} + x^\top(0)(Q + \Omega^\top \bar{Q}\Omega)x(0)$$

subject to

$$[A_x \ A_x \ \cdots \ A_x](\Omega x(0) + \Gamma \boldsymbol{u}) \leq [b_x \ b_x \ \ldots \ b_x]^T, ,$$

$$[A_u \ A_u \ \cdots \ A_u]\boldsymbol{u} \leq [b_u \ b_u \ \ldots \ b_u]^T,$$

where the initial state in the prediction model is set equal to $x(t)$.

In Matlab

```
[x,fval,exitflag,output]=quadprog(H,f,A,b,[],[],lb,ub,[])
```

Motion planning. The optimization problem

If the system is linear ($x^+ = Ax + Bu$), the cost quadratic and the constraints convex

$$\mathcal{X} = \{x : \; A_x \leq b_x\}, \; \mathcal{U} = \{u : \; A_u \leq b_u\}.$$

The constrained optimization problem becomes (see the batch approach for LQ)

$$\min_{\boldsymbol{u}} \boldsymbol{u}^\top (\Gamma^\top \bar{Q} \Gamma + \bar{R}) \boldsymbol{u} + 2x^\top(0) \Omega^\top \bar{Q} \Gamma \boldsymbol{u} + x^\top(0)(Q + \Omega^\top \bar{Q} \Omega) x(0)$$

subject to

$$[A_x \; A_x \; \cdots \; A_x](\Omega x(0) + \Gamma \boldsymbol{u}) \leq [b_x \; b_x \; \ldots \; b_x]^T, ,$$

$$[A_u \; A_u \; \cdots \; A_u] \boldsymbol{u} \leq [b_u \; b_u \; \ldots \; b_u]^T,$$

where the initial state in the prediction model is set equal to $x(t)$.

In Matlab

```
[x,fval,exitflag,output]=quadprog(H,f,A,b,[],[],lb,ub,[])
```

Apply to the system the control input $u(t) = u^*(0)$ and repeat the optimization from the next state (over a shifted time horizon).

Optimization-based motion planning problem formulation

The optimization-based motion planning reads as

$$
\begin{aligned}
\underset{\textit{path/trajectory}}{\text{minimize}} \quad & \text{deviation from reference path/trajectory} \\
\text{subject to} \quad & \\
& \text{vehicle dynamics} \\
& \text{safety constraints} \\
& \text{actuator limitations} \\
& \text{design (e.g., comfort) constraints}
\end{aligned}
$$

The planned motion is periodically updated based on the current vehicle and environment state, and sent to motion control layer.

Space vs. Time-domain problem formulations

We have introduced vehicle models in the form

$$\dot{x} = f(x, u), \tag{3}$$

where the state and inputs are functions of time.

Space vs. Time-domain problem formulations

We have introduced vehicle models in the form

$$\dot{x} = f(x, u), \tag{3}$$

where the state and inputs are functions of time.

The vehicle model can be rewritten as

$$\frac{dx}{ds} =$$

Space vs. Time-domain problem formulations

We have introduced vehicle models in the form

$$\dot{x} = f(x, u), \tag{3}$$

where the state and inputs are functions of time.

The vehicle model can be rewritten as

$$\frac{dx}{ds} = \frac{dt}{ds}\frac{dx}{dt}$$

Space vs. Time-domain problem formulations

We have introduced vehicle models in the form

$$\dot{x} = f(x, u), \tag{3}$$

where the state and inputs are functions of time.

The vehicle model can be rewritten as

$$\frac{dx}{ds} = \frac{dt}{ds}\frac{dx}{dt} = \frac{1}{V_s} f(x(s), u(s)), \tag{4}$$

where s and V_s are the traveled distance and the vehicle speed over the path.

Time-domain. With (3) as prediction model, the problem is formulated over a finite-time horizon.

Space vs. Time-domain problem formulations

We have introduced vehicle models in the form

$$\dot{x} = f(x, u), \tag{3}$$

where the state and inputs are functions of time.

The vehicle model can be rewritten as

$$\frac{dx}{ds} = \frac{dt}{ds}\frac{dx}{dt} = \frac{1}{V_s} f(x(s), u(s)), \tag{4}$$

where s and V_s are the traveled distance and the vehicle speed over the path.

Time-domain. With (3) as prediction model, the problem is formulated over a finite-time horizon. Cost, reference and constraints needs to be defined w.r.t. the time variable.

Space vs. Time-domain problem formulations

We have introduced vehicle models in the form

$$\dot{x} = f(x, u), \tag{3}$$

where the state and inputs are functions of time.

The vehicle model can be rewritten as

$$\frac{dx}{ds} = \frac{dt}{ds}\frac{dx}{dt} = \frac{1}{V_s} f(x(s), u(s)), \tag{4}$$

where s and V_s are the traveled distance and the vehicle speed over the path.

Time-domain. With (3) as prediction model, the problem is formulated over a finite-time horizon. Cost, reference and constraints needs to be defined w.r.t. the time variable.

Space-domain. With (3) as prediction model, the problem is formulated over a finite-distance ahead.

Space vs. Time-domain problem formulations

We have introduced vehicle models in the form

$$\dot{x} = f(x, u), \tag{3}$$

where the state and inputs are functions of time.

The vehicle model can be rewritten as

$$\frac{dx}{ds} = \frac{dt}{ds}\frac{dx}{dt} = \frac{1}{V_s} f(x(s), u(s)), \tag{4}$$

where s and V_s are the traveled distance and the vehicle speed over the path.

Time-domain. With (3) as prediction model, the problem is formulated over a finite-time horizon. Cost, reference and constraints needs to be defined w.r.t. the time variable.

Space-domain. With (3) as prediction model, the problem is formulated over a finite-distance ahead. Cost, reference and constraints needs to be defined w.r.t. the traveled distance s.

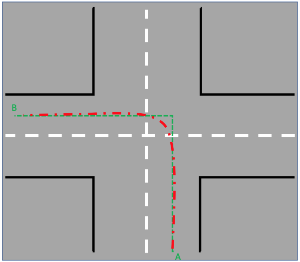
Cost function

The *motion planning objective* is planning a path that follows a reference path to the extent allowed by the vehicle physical limitations and the obstacles.

Cost function

The *motion planning objective* is planning a path that follows a reference path to the extent allowed by the vehicle physical limitations and the obstacles.

Assume a reference path x^r is given along with a corresponding input trajectory u^r.

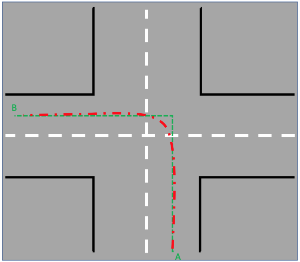


Cost function

The *motion planning objective* is planning a path that follows a reference path to the extent allowed by the vehicle physical limitations and the obstacles.

Assume a reference path x^r is given along with a corresponding input trajectory u^r.

Define the tracking errors $e_x = x - x^r$, $e_u = u = u^r$.



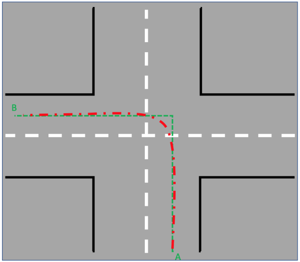
Quadratic cost. Squared, 2-norm of the distance from the path

$$V_N(x(0), u(0:N-1)) = e_x(N)^\top P_f e_x(N) + \sum_{i=0}^{N-1} e_x^\top(i) Q e_x(N)(i) + e_u(i)^\top(i) R e_u(i).$$

Cost function

The *motion planning objective* is planning a path that follows a reference path to the extent allowed by the vehicle physical limitations and the obstacles.

Assume a reference path x^r is given along with a corresponding input trajectory u^r.



Define the tracking errors $e_x = x - x^r$, $e_u = u = u^r$.

Quadratic cost. Squared, 2-norm of the distance from the path

$$V_N(x(0), u(0{:}N-1)) = e_x(N)^\top P_f e_x(N) + \sum_{i=0}^{N-1} e_x^\top(i) Q e_x(N)(i) + e_u(i)^\top(i) R e_u(i).$$

Strongly penalizes large deviation from the path.

Cost function

Linear cost. ∞-norm of the distance from the path.

Cost function

Linear cost. ∞-norm of the distance from the path. Recall that

$$\|x\|_\infty = \max_i |x_i|.$$

Cost function

Linear cost. ∞-norm of the distance from the path. Recall that

$$\|x\|_\infty = \max_i |x_i|.$$

The cost

$$V_N(x(0), u(0\!:\!N-1)) = \|P_f e_x(N)\|_\infty + \sum_{i=0}^{N-1} \|Q e_x(i)\|_\infty + \|R e_u(i)\|_\infty$$

penalizes the maximum deviation from the path over the horizon.

Cost function

Linear cost. ∞-norm of the distance from the path. Recall that

$$\|x\|_\infty = \max_i |x_i|.$$

The cost

$$V_N(x(0), u(0{:}N-1)) = \|P_f e_x(N)\|_\infty + \sum_{i=0}^{N-1} \|Q e_x(i)\|_\infty + \|R e_u(i)\|_\infty$$

penalizes the maximum deviation from the path over the horizon.

The minimization of V_N results into a linear cost.

Reference path

The reference path can be expressed as

- ***Waypoints.*** A n-tuple of (x, y, ψ) poses.

Reference path

The reference path can be expressed as

- ***Waypoints.*** A n-tuple of (x, y, ψ) poses. ***Pro:*** Simple, most common way to define a path.

Reference path

The reference path can be expressed as

- ***Waypoints.*** A n-tuple of (x, y, ψ) poses. ***Pro:*** Simple, most common way to define a path. ***Cons:*** Curvature must be imposed in the generation of the poses sequence.

Reference path

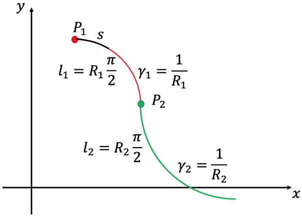
The reference path can be expressed as

- ***Waypoints.*** A n-tuple of (x, y, ψ) poses. ***Pro:*** Simple, most common way to define a path. ***Cons:*** Curvature must be imposed in the generation of the poses sequence.
- ***Composition of curves.*** E.g., straight segments,

Reference path

The reference path can be expressed as

- ***Waypoints.*** A n-tuple of (x, y, ψ) poses. ***Pro:*** Simple, most common way to define a path. ***Cons:*** Curvature must be imposed in the generation of the poses sequence.
- ***Composition of curves.*** E.g., straight segments,
  - constant curvature arcs,



$$\begin{aligned} x(s) &= P_1^x + R_1 \sin \tfrac{s}{R_1}, \\ y(s) &= P_1^y - R_1\left(1 - \cos \tfrac{s}{R_1}\right), && 0 \le s \le l_1 \\ \psi(s) &= -\tfrac{s}{R_1}, \end{aligned}$$

$$\begin{aligned} x(s) &= P_2^x + R_2\left(1 - \cos \tfrac{s-l_1}{R_2}\right), \\ y(s) &= P_2^y + R_2 \sin \tfrac{s-l_1}{R_2}, && l_1 \le s \le l_2 \\ \psi(s) &= -\tfrac{\pi}{2} + \tfrac{s-l_1}{R_1}, \end{aligned}$$

Reference path

The reference path can be expressed as

- ***Composition of curves.*** E.g., straight segments, constant curvature arcs,

Reference path

The reference path can be expressed as

- ***Composition of curves.*** E.g., straight segments, constant curvature arcs,
  - clothoids (linearly increasing curvature). *Fresnel integrals* can be used to calculate the pose

$$x(s) = \int_0^s \sin \tau^2 d\tau, \quad y(s) = \int_0^s \cos \tau^2 d\tau, \quad \psi(s) = s^2.$$

Reference path

The reference path can be expressed as

- ***Composition of curves.*** E.g., straight segments, constant curvature arcs,
  - clothoids (linearly increasing curvature). *Fresnel integrals* can be used to calculate the pose

$$x(s) = \int_0^s \sin \tau^2 d\tau, \quad y(s) = \int_0^s \cos \tau^2 d\tau, \quad \psi(s) = s^2.$$

The curvature is $\rho = 2s$

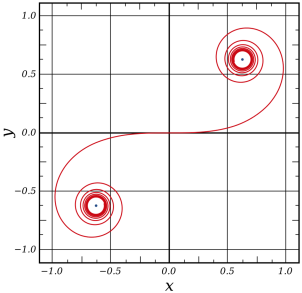
Reference path

The reference path can be expressed as

- ***Composition of curves.*** E.g., straight segments, constant curvature arcs,
    - clothoids (linearly increasing curvature). *Fresnel integrals* can be used to calculate the pose

$$x(s) = \int_0^s \sin \tau^2 d\tau, \quad y(s) = \int_0^s \cos \tau^2 d\tau, \quad \psi(s) = s^2.$$

The curvature is $\rho = 2s$ and the resulting curve is



Path sampling

In order to build the cost function in an MPC-based motion planning problem the reference pose needs to be provided *(i)* at specific points $s_1, \ldots, s_N$ along the path or *(ii)* time instants $t_1, \ldots, t_N$.

Path sampling

In order to build the cost function in an MPC-based motion planning problem the reference pose needs to be provided *(i)* at specific points $s_1, \ldots, s_N$ along the path or *(ii)* time instants $t_1, \ldots, t_N$.

In case *(i)*, we distinguish two cases

Path sampling

In order to build the cost function in an MPC-based motion planning problem the reference pose needs to be provided *(i)* at specific points $s_1, \dots, s_N$ along the path or *(ii)* time instants $t_1, \dots, t_N$.

In case *(i)*, we distinguish two cases

1. The reference path is given as sequence of waypoints. In this case any interpolation method does the job of finding the path in between the waypoints.

Path sampling

In order to build the cost function in an MPC-based motion planning problem the reference pose needs to be provided *(i)* at specific points $s_1, \ldots, s_N$ along the path or *(ii)* time instants $t_1, \ldots, t_N$.

In case *(i)*, we distinguish two cases

1. The reference path is given as sequence of waypoints. In this case any interpolation method does the job of finding the path in between the waypoints.
2. The reference path is given as a composition of curves. It is enough the evaluate the curve at the specific values of s

Path sampling

In order to build the cost function in an MPC-based motion planning problem the reference pose needs to be provided *(i)* at specific points $s_1, \ldots, s_N$ along the path or *(ii)* time instants $t_1, \ldots, t_N$.

In case *(i)*, we distinguish two cases

1. The reference path is given as sequence of waypoints. In this case any interpolation method does the job of finding the path in between the waypoints.
2. The reference path is given as a composition of curves. It is enough the evaluate the curve at the specific values of s

In case *(ii)*, the path is parametrized w.r.t. the time

Path sampling

In order to build the cost function in an MPC-based motion planning problem the reference pose needs to be provided *(i)* at specific points $s_1, \ldots, s_N$ along the path or *(ii)* time instants $t_1, \ldots, t_N$.

In case *(i)*, we distinguish two cases

1. The reference path is given as sequence of waypoints. In this case any interpolation method does the job of finding the path in between the waypoints.
2. The reference path is given as a composition of curves. It is enough the evaluate the curve at the specific values of s

In case *(ii)*, the path is parametrized w.r.t. the time

$$x^r = x^r(t), \; u^r = u^r(t).$$

Path sampling

In order to build the cost function in an MPC-based motion planning problem the reference pose needs to be provided *(i)* at specific points $s_1, \dots, s_N$ along the path or *(ii)* time instants $t_1, \dots, t_N$.

In case *(i)*, we distinguish two cases

1. The reference path is given as sequence of waypoints. In this case any interpolation method does the job of finding the path in between the waypoints.
2. The reference path is given as a composition of curves. It is enough the evaluate the curve at the specific values of s

In case *(ii)*, the path is parametrized w.r.t. the time

$$x^r = x^r(t),\ u^r = u^r(t).$$

In this case the prediction model can be augmented with the state (time dynamics)

$$\tau^+ = \tau + T_s + v.$$

Path sampling

In order to build the cost function in an MPC-based motion planning problem the reference pose needs to be provided *(i)* at specific points $s_1, \ldots, s_N$ along the path or *(ii)* time instants $t_1, \ldots, t_N$.

In case *(i)*, we distinguish two cases

1. The reference path is given as sequence of waypoints. In this case any interpolation method does the job of finding the path in between the waypoints.
2. The reference path is given as a composition of curves. It is enough the evaluate the curve at the specific values of s

In case *(ii)*, the path is parametrized w.r.t. the time

$$x^r = x^r(t),\ u^r = u^r(t).$$

In this case the prediction model can be augmented with the state (time dynamics)

$$\tau^+ = \tau + T_s + v.$$

The additional control input v is used to avoid aggressive maneuvers due to obstacles that may lead to large tracking errors.

Vehicle modeling

Usually a simple vehicle model is used for motion planning.

Vehicle modeling

Usually a simple vehicle model is used for motion planning. At rater low speed a kinematic model can be used

$$
\begin{aligned}
\dot{X} &= V\cos(\psi + \beta), \\
\dot{Y} &= V\sin(\psi + \beta), \\
\dot{\psi} &= \frac{V\cos\beta}{l_f + l_r}\left(\tan\delta_f - \tan\delta_r\right), \\
\beta &= \tan^{-1}\left(\frac{l_f\tan\delta_r + l_r\tan\delta_f}{l_f + l_r}\right)
\end{aligned}
$$

Vehicle modeling

Usually a simple vehicle model is used for motion planning. At rater low speed a kinematic model can be used

$$
\begin{aligned}
\dot{X} &= V\cos(\psi+\beta),\\
\dot{Y} &= V\sin(\psi+\beta),\\
\dot{\psi} &= \frac{V\cos\beta}{l_f+l_r}\left(\tan\delta_f - \tan\delta_r\right),\\
\beta &= \tan^{-1}\left(\frac{l_f\tan\delta_r + l_r\tan\delta_f}{l_f+l_r}\right)
\end{aligned}
$$

This is to be discretized in either the time or space domain.

Vehicle modeling

Usually a simple vehicle model is used for motion planning. At rater low speed a kinematic model can be used

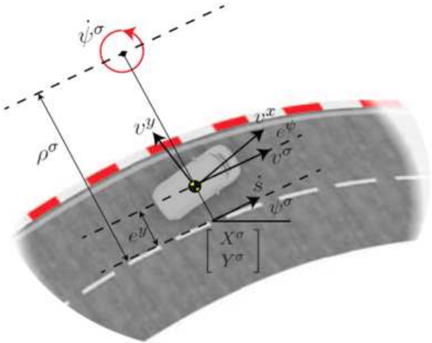
$$
\begin{aligned}
\dot{X} &= V\cos(\psi+\beta),\\
\dot{Y} &= V\sin(\psi+\beta),\\
\dot{\psi} &= \frac{V\cos\beta}{l_f+l_r}\left(\tan\delta_f-\tan\delta_r\right),\\
\beta &= \tan^{-1}\left(\frac{l_f\tan\delta_r+l_r\tan\delta_f}{l_f+l_r}\right)
\end{aligned}
$$

This is to be discretized in either the time or space domain.

In case a bicycle model is used to plan the motion, motion planning and control can be lumped together into a single task.

Vehicle modeling

Let's rewrite the kinematic model in the space domain, w.r.t. a reference path $(x^{\sigma}(s), y^{\sigma}\sigma(s))$.

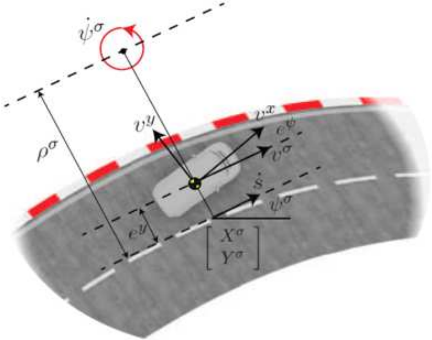


Vehicle modeling

Let's rewrite the kinematic model in the space domain, w.r.t. a reference path $(x^\sigma(s), y^\sigma\sigma(s))$.

Define the deviation and orientation error w.r.t. the path in terms of the vehicle pose in the global frame

$$
\begin{aligned}
e_y &= \cos\psi^\sigma(Y - y^\sigma) - \sin\psi^r(X - x^\sigma),\\
e_\psi &= \psi - \psi^\sigma.
\end{aligned}
$$



Vehicle modeling

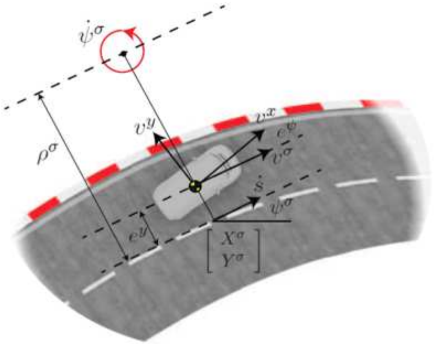
Let's rewrite the kinematic model in the space domain, w.r.t. a reference path $(x^\sigma(s), y^\sigma\sigma(s))$.

Define the deviation and orientation error w.r.t. the path in terms of the vehicle pose in the global frame

$$
\begin{aligned}
e_y &= \cos\psi^\sigma(Y - y^\sigma) - \sin\psi^r(X - x^\sigma),\\
e_\psi &= \psi - \psi^\sigma.
\end{aligned}
$$

The component v^σ of the vehicle speed parallel to the path

$$
\begin{aligned}
v^\sigma &= (\rho^\sigma - e_y)\dot{\psi}^r,\\
v^\sigma &= V_x \cos e_\psi - V_y \sin e_\psi.
\end{aligned}
$$



Vehicle modeling

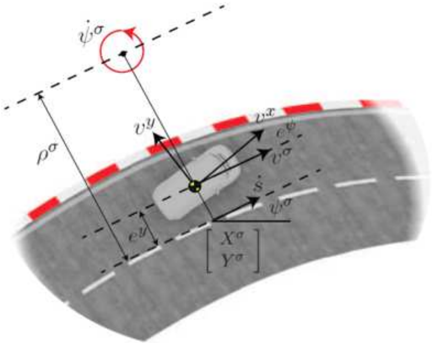
Let's rewrite the kinematic model in the space domain, w.r.t. a reference path $(x^\sigma(s), y^\sigma\sigma(s))$.

Define the deviation and orientation error w.r.t. the path in terms of the vehicle pose in the global frame

$$
\begin{aligned}
e_y &= \cos\psi^\sigma(Y - y^\sigma) - \sin\psi^r(X - x^\sigma),\\
e_\psi &= \psi - \psi^\sigma.
\end{aligned}
$$

The component v^σ of the vehicle speed parallel to the path

$$
\begin{aligned}
v^\sigma &= (\rho^\sigma - e_y)\dot{\psi}^r,\\
v^\sigma &= V_x\cos e_\psi - V_y\sin e_\psi.
\end{aligned}
$$



gives $V_s = \dot{s} = \rho^\sigma\dot{\psi}^\sigma = \frac{1}{1-\frac{e_y}{\rho^\sigma}}\left(V_x\cos e_\psi - V_y\sin e_\psi\right),$

Vehicle modeling

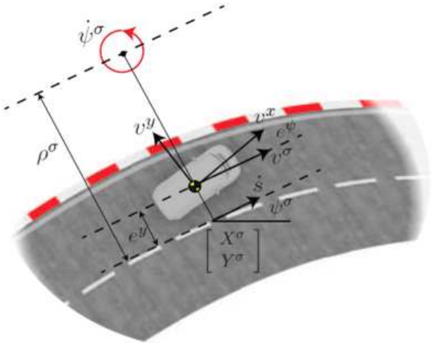
Let's rewrite the kinematic model in the space domain, w.r.t. a reference path $(x^\sigma(s), y^\sigma\sigma(s))$.

Define the deviation and orientation error w.r.t. the path in terms of the vehicle pose in the global frame

$$
\begin{aligned}
e_y &= \cos\psi^\sigma(Y - y^\sigma) - \sin\psi^r(X - x^\sigma), \\
e_\psi &= \psi - \psi^\sigma.
\end{aligned}
$$

The component v^σ of the vehicle speed parallel to the path

$$
\begin{aligned}
v^\sigma &= (\rho^\sigma - e_y)\dot{\psi}^r, \\
v^\sigma &= V_x \cos e_\psi - V_y \sin e_\psi.
\end{aligned}
$$



gives $V_s = \dot{s} = \rho^\sigma\dot{\psi}^\sigma = \frac{1}{1-\frac{e_y}{\rho^\sigma}}\left(V_x \cos e_\psi - V_y \sin e_\psi\right)$, where $V_x,\ V_y$ can be expressed in terms of the vehicle pose in the global frame.

Vehicle modeling

The kinematic model in the space domain is finally written as

Vehicle modeling

The kinematic model in the space domain is finally written as

$$
\begin{aligned}
\frac{de_y}{ds} = e_y' &= \frac{1}{V_s}\left(V_x \sin e_\psi + V_x \frac{l_r}{l_f + l_r}\delta \cos e_\psi\right), \\
e_\psi' &= \frac{V_x \delta}{V_s(l_f + l_r)} - \frac{1}{R(s)}, \\
V_x' &= \frac{\dot{V}_x}{V_s},
\end{aligned}
$$

Vehicle modeling

The kinematic model in the space domain is finally written as

$$\frac{de_y}{ds} = e_y' = \frac{1}{V_s}\left(V_x \sin e_\psi + V_x \frac{l_r}{l_f + l_r}\delta \cos e_\psi\right),$$
$$e_\psi' = \frac{V_x \delta}{V_s(l_f + l_r)} - \frac{1}{R(s)},$$
$$V_x' = \frac{\dot{V}_x}{V_s},$$

where $R(s)$ is the curvature radius of the reference path, $\dot{V}_x$ is the commanded longitudinal acceleration (control input).

Vehicle modeling

The kinematic model in the space domain is finally written as

$$\begin{aligned}
\frac{de_y}{ds} = e_y' &= \frac{1}{V_s}\left(V_x \sin e_\psi + V_x \frac{l_r}{l_f + l_r}\delta \cos e_\psi\right), \\
e_\psi' &= \frac{V_x \delta}{V_s(l_f + l_r)} - \frac{1}{R(s)}, \\
V_x' &= \frac{\dot{V}_x}{V_s},
\end{aligned}$$

where $R(s)$ is the curvature radius of the reference path, $\dot{V}_x$ is the commanded longitudinal acceleration (control input).

Note that, all variables are to be expressed in the space domain (they are function of s).

Safety constraints. Static obstacles

By rewriting the vehicle model in the space domain, where the coordinates describe the vehicle position and orientation w.r.t. the reference path, the safety constraints simply become

$$e_y(s) \in \left[-\frac{L_w(s)}{2},\ \frac{L_w(s)}{s}\right],$$

where $L_w(s)$ is the lane width at s.

Safety constraints. Static obstacles

By rewriting the vehicle model in the space domain, where the coordinates describe the vehicle position and orientation w.r.t. the reference path, the safety constraints simply become

$$e_y(s) \in \left[-\frac{L_w(s)}{2},\ \frac{L_w(s)}{s}\right],$$

where $L_w(s)$ is the lane width at s.

Assuming the position within the lane of *static obstacles* is provided by a sensing system,

$$e_y^{obs}(s) \in \mathcal{X}^{obs} = \left[e_y^{obs,min},\ e_y^{obs,max}\right],\ s \in [s^{obs,min},\ s^{obs,max}]$$

Safety constraints. Static obstacles

By rewriting the vehicle model in the space domain, where the coordinates describe the vehicle position and orientation w.r.t. the reference path, the safety constraints simply become

$$e_y(s) \in \left[-\frac{L_w(s)}{2},\ \frac{L_w(s)}{s}\right],$$

where $L_w(s)$ is the lane width at s.

Assuming the position within the lane of *static obstacles* is provided by a sensing system,

$$e_y^{obs}(s) \in \mathcal{X}^{obs} = \left[e_y^{obs,min},\ e_y^{obs,max}\right],\ s \in [s^{obs,min},\ s^{obs,max}]$$

collision avoidance constraints are imposed by

$$e_y(s) \in \left[-\frac{3L_w(s)}{2},\ \frac{L_w(s)}{s}\right] \setminus \mathcal{X}^{obs},\ s \in [s^{obs,min},\ s^{obs,max}]$$